On the Heterogeneity Bias of Cost Matrices for Assessing Scheduling Algorithms

Louis-Claude Canon, Laurent Philippe

On the Heterogeneity Bias of Cost Matrices for Assessing Scheduling Algorithms

Louis-Claude Canon and Laurent Philippe

Abstract—Assessing the performance of scheduling heuristics through simulation requires one to generate synthetic instances of tasks and machines with well-identified properties. Carefully controlling these properties is mandatory to avoid any bias. We consider the scheduling problem consisting of allocating independent sequential tasks on unrelated machines while minimizing the maximum execution time. In this problem, the instance is a cost matrix that specifies the execution cost of any task on any machine. This article proposes two measures for quantifying the heterogeneity properties of a cost matrix. An analysis of two classical methods used in the literature reveals a bias in previous studies. We propose new methods to generate instances with given heterogeneity properties and we show that heterogeneity has a significant impact on twelve heuristics.

Index Terms—scheduling, cost matrix, heterogeneity, bias, parallelism, unrelated, measure

1 Introduction

Leveraging the parallelism of multi-core distributed platforms involves efficiently scheduling applications on several machines [1]. Current studies rely on performance evaluation to determine the best solution for any underlying problem. This process can be divided into distinct categories: formal analysis, experiments, simulations, etc. In the case of simulations, a scheduling strategy is tested in a virtual environment with a given workload. This paper focuses on the generation of synthetic instances.

Synthetic instances of workload allow a more general evaluation than with specific traces. They are particularly useful for sensitivity analysis [2], which consists in assessing the impact of the instance properties on the algorithms. However, the lack of control on the instance properties makes it difficult to confront the results of independent studies. For instance, although many papers have compared several scheduling heuristics [3], [4], [5], [6], predicting their performance is still an issue. These problems can be tackled by carefully controlling the instance properties.

Specifically, we consider the scheduling problem noted $R||C_{\max}$ in $\alpha|\beta|\gamma$ notation [7]. It consists in scheduling n independent sequential tasks on m unrelated machines. All tasks are available simultaneously and preemption is not possible. The instance is a *cost matrix* where each element $e_{i,j}$ is a positive integer that represents the execution cost of task i on machine j. The objective is to allocate each task to a machine such that the maximum execution time on any machine is minimized. More formally, we want to minimize $\max(\sum \pi(i,j) \times e_{i,j})$ where $\pi(i,j)$ is equal to one if task i is scheduled on machine j and zero otherwise.

This problem corresponds to numerous practical situations where a set of tasks, either identical or heterogeneous, must be distributed on platforms ranging from grids to homogeneous clusters and including semi-heterogeneous platforms such as CPU/GPU platforms. This is the case of a master/slave application that is publicly distributed. To efficiently run on several platforms the master must include a component that chooses where to run each task. The choice of the scheduling algorithm is a key point for the software performance.

To reflect the diversity of heterogeneous platforms, a fair comparison of scheduling heuristics must rely on a set of cost matrices that have distinct properties. Controlling the generation of synthetic random cost matrix in this context enables an assessment on a panel of instances that is sufficiently large to encompass practical settings that are currently existing or yet to come. In this generation, it is therefore crucial to identify and control the properties that impact the most critically the performance such as the heterogeneity.

For this problem, the range-based and CVB (Coefficient of Variation Based) methods proposed in [8], [9] are currently the standard methods used in the literature to generate instances. However, the properties of the matrices generated with these methods have never been formally analyzed and previous studies may thus be exposed to a bias.

This paper provides the following contributions:[1]

- a statistical description of the use of the range-based and CVB methods in the literature (Section 3);
- a study of how to quantify the heterogeneity properties of a cost matrix (Section 4);
- a formal analysis of the range-based and CVB methods and the identification of a bias that impacts several studies (Section 4);
- a new method with control over heterogeneity properties (Section 5);
- and, an assessment of the impact of these properties on twelve heuristics (Section 6).

2 Related Work

The concept of heterogeneity was first introduced in the context of cost matrix by Armstrong [13]. He described the

• *L.-C. Canon and L. Philippe are with FEMTO-ST / CNRS / UBFC and the Université de Franche-Comté, Besançon, France. E-mail: {louis-claude.canon,laurent.philippe}@univ-fcomte.fr*

1. The related code, data and analysis are available in [10]. Most of these results are also available in the companion research report [11] and in a conference paper [12].

heterogeneity quadrant in which cost matrices are divided into four categories depending on their heterogeneity properties regarding tasks and machines: low/low, low/high, high/low, and high/high. For instance low/high refers to low task heterogeneity and high machine heterogeneity. However, no method for generating such matrices was proposed.

The range-based and CVB methods were first proposed to fill this gap in [14] and then in [8], [9]. However, task and machine heterogeneities were not formally defined and analyzed. The methods were assumed to generate matrices with the expected properties and only validated through some examples.

The limits of these methods were later acknowledged in [15], which proposed to consider the average coefficient of variation[2], skewness and kurtosis of the costs for each task and for each machine. The proposed scheme (based on decision trees) uses these additional information to predict scheduling heuristic performance. Despite a wide experimentation plan, the study lacks discussion and interpretation in particular on the relative importance of the considered measures. Additionally, no formal analysis was provided. The exhibited decision trees suggest that the average coefficient of variation plays a significant role and our proposed measures rely on this coefficient.

The MPH (Machine Performance Homogeneity) is introduced in [16] for capturing the heterogeneity between the machines while its counterpart for the tasks, the TDH (Task Difficulty Homogeneity), appears in [17]. We discuss them more extensively in Section 4. In addition, the TMA (Task-Machine Affinity) is also defined in [16]: it quantifies the specialisation of the system (i.e., whether some machines are particularly efficient for some specific tasks). Although the three measures are applied to a real benchmark, no method is proposed for generating matrices with given MPH, TDH and TMA. It is thus unclear what is the impact of the proposed measures on heuristic performance. Finally, they show that the range-based and CVB methods do not cover the entire range of possible values for the MPH and the TMA, which is consistent with the conclusion of Section 4.

Friese et al. [18] present a method for adding tasks in a given cost matrix while preserving some statistical properties on the costs of each machine (mean, coefficient of variation, skewness and kurtosis). It ignores the properties of the costs of each task however.

A method for generating matrices with varying affinities (similar to the TMA) is proposed in [19]. It is similar to the noise-based method described in Section 5, but no formal analysis is provided.

Khemka et al. [20] propose a method for changing the TMA of an existing matrix while keeping the same MPH and TDH. TMA is mentioned to be related to the correlation. Investigating the correlation properties is left for future work. There is also another body of literature dedicated to the generation of matrices with given correlation and covariance matrices [21].

Finally, the problem of generating contingency tables is close to our problem. The objective is to generate a uniform matrix with given row and column sums (we consider average in our problem instead). One significant approach consists in using Markov chain Monte Carlo (MCMC) methods [22].

2. Ratio of the standard deviation to the mean.

ALGORITHM 1: Range-based cost matrix generation with the uniform distribution

Input: n, m, R_{task}, R_{mach}
Output: a $n \times m$ cost matrix
1: **for all** $1 \leq i \leq n$ **do** {Generate each row}
2: $\quad \tau[i] \leftarrow U(1, R_{\text{task}})$
3: $\quad$ **for all** $1 \leq j \leq m$ **do** {Generate each value of the row}
4: $\quad\quad e_{i,j} \leftarrow \tau[i] \times U(1, R_{\text{mach}})$
5: $\quad$ **end for**
6: **end for**
7: **return** $\{e_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$

However, when used directly, such methods introduce a large variance in the costs, which hides the effect of the heterogeneity. The shuffling method we introduce below has similarities with MCMC methods but limits the introduced variance.

3 Matrix Generation Methods

The most used methods for generating cost matrices are the range-based and the CVB (Coefficient of Variation Based) methods [8], [9], [14]. Table 1 summarizes the most frequent notations.

Table 1: List of notations

Symbol	Definition
i	index of the tasks
j	index of the machines
n	number of tasks
m	number of machines
$e_{i,j}$	execution cost of task i on machine j
w_i	weight of task i
b_j	inverse speed of machine j
$U(A, B)$	uniform distribution between A and B
$G(\alpha, \beta)$	gamma distribution with shape α and scale β
R_{task}	parameter for the range-based method
R_{mach}	parameter for the range-based method
V_{task}	parameter for the CVB, shuffling and noise-based methods
V_{mach}	parameter for the CVB, shuffling and noise-based methods
V_{noise}	parameter for the noise-based method
a	fraction of the consistent rows
b	fraction of the consistent columns
$V\mu_{\text{task}}$	first measure of task heterogeneity
$V\mu_{\text{mach}}$	first measure of machine heterogeneity
μV_{task}	second measure of task heterogeneity
μV_{mach}	second measure of machine heterogeneity

3.1 Range-Based Method

The range-based method generates n vectors of m values that follow a uniform distribution in the range $[1, R_{\text{mach}}]$ (see Algorithm 1). Each row is then multiplied by a random value that follows a uniform distribution in the range $[1, R_{\text{task}}]$ (Line 2). The resulting cost matrix is similar to the following (where τ is a vector of n uniform values in $[1, R_{\text{task}}]$):

$$\begin{pmatrix} \tau[1]U(1, R_{\text{mach}}) & \cdots & \tau[1]U(1, R_{\text{mach}}) \\ \vdots & \ddots & \vdots \\ \tau[n]U(1, R_{\text{mach}}) & \cdots & \tau[n]U(1, R_{\text{mach}}) \end{pmatrix}$$

ALGORITHM 2: CVB cost matrix generation with the gamma distribution

Input: n, m, V_{task}, V_{mach}, μ_{task}
Output: a $n \times m$ cost matrix
1: $\alpha_{\text{task}} \leftarrow 1/V_{\text{task}}^2$
2: $\alpha_{\text{mach}} \leftarrow 1/V_{\text{mach}}^2$
3: $\beta_{\text{task}} \leftarrow \mu_{\text{task}}/\alpha_{\text{task}}$
4: **for all** $1 \leq i \leq n$ **do**
5: $\quad q[i] \leftarrow G(\alpha_{\text{task}}, \beta_{\text{task}})$
6: $\quad \beta_{\text{mach}}[i] \leftarrow q[i]/\alpha_{\text{mach}}$
7: $\quad$ **for all** $1 \leq j \leq m$ **do**
8: $\quad\quad e_{i,j} \leftarrow G(\alpha_{\text{mach}}, \beta_{\text{mach}}[i])$
9: $\quad$ **end for**
10: **end for**
11: **return** $\{e_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$

Proposition 1. *When used with parameters R_{task} and R_{mach}, the range-based method generates costs with expected value $\frac{1}{4}(R_{task}+1)(R_{mach}+1)$ and standard deviation $\frac{1}{12}[(R_{task}-1)^2(R_{mach}-1)^2 + 3(R_{mach}-1)^2(R_{task}+1)^2 + 3(R_{task}-1)^2(R_{mach}+1)^2]^{1/2}$.*

Proof. Each cost is the product of $\tau[i]$, which follows a uniform law in the range $[1, R_{\text{task}}]$, and a random variable that follows a uniform law in the range $[1, R_{\text{mach}}]$. Therefore, the expected value of the costs is the product of the expected values of both distributions, namely $(R_{\text{task}}+1)/2$ and $(R_{\text{mach}}+1)/2$.

The standard deviation of the product of two random variables with means μ_1 and μ_2, and standard deviations σ_1 and σ_2 is $\sqrt{\sigma_1^2\sigma_2^2 + \mu_1^2\sigma_2^2 + \sigma_1^2\mu_2^2}$. With a similar argument as for the expected value, we can derive the standard deviation of the costs. □

Table 2 summarizes the properties of this method. Except for low values of R_{task} and R_{mach}, the CV (Coefficient of Variation) remains close to a constant. For instance, when $R_{\text{task}} = R_{\text{mach}} = 100$, then the CV is around 0.86. As shown in Section 4, this method is not well-suited to control the heterogeneity of the resulting cost matrix. Also, given that this method is asymmetric, it may be expected to handle task heterogeneity differently from machine heterogeneity.

3.2 CVB Method

The CVB method is based on the same principle except it uses parameters that are distinct from the underlying distribution parameters. In particular, it requires two CV (V_{task} for the tasks and V_{mach} for the machines) and one mean (μ_{task} for the tasks). The random values follow a gamma distribution whose parameters are computed such that the provided CV and mean are respected.

Proposition 2. *When used with parameters V_{task}, V_{mach} and μ_{task}, the CVB method generates costs with expected value μ_{task} and coefficient of variation $\sqrt{V_{task}^2 V_{mach}^2 + V_{task}^2 + V_{mach}^2}$.*

Proof. In order to apply the same analysis as in the proof of Proposition 1, we need to prove that any cost is the product of two gamma distributions. More precisely, we need to prove that the random generation on Line 8 is equivalent to multiplying $q[i]$ by a gamma law with mean one and CV V_{mach}.

Each cost $e_{i,j}$ is a random variable that follows a gamma distribution with mean $q[i]$ and CV V_{mach}. The probability that $e_{i,j}$ is no more than x is given by $\frac{1}{\Gamma(\alpha)}\gamma(\alpha, \frac{x}{\beta})$ where $\alpha = 1/V_{\text{mach}}^2$, $\beta = q[i]/\alpha$, $\Gamma(\alpha)$ is the gamma function and $\Gamma(\alpha, \frac{x}{\beta})$ is the lower incomplete gamma function.

By contrast, let X be a random variable that follows a gamma distribution with mean one and CV V_{mach}. Then, the probability that $q[i]X$ is no more than x is the probability that X is no more than $x/q[i]$: $\frac{1}{\Gamma(\alpha)}\gamma(\alpha, \frac{x/q[i]}{\beta})$ where $\alpha = 1/V_{\text{mach}}^2$ and $\beta = 1/\alpha$. It is thus the same as for $e_{i,j}$.

Thus, Line 8 can be replaced by the product of $q[i]$ by a gamma law with mean one and CV V_{mach} (i.e., $e_{i,j} \leftarrow q[i]G(\alpha_{\text{mach}}, 1/\alpha_{\text{mach}})$), which is the product of two gamma distributions.

The proof is then analogous to the proof of Proposition 1. □

Table 3 summarizes the properties of this method, which is more adapted to control the heterogeneity of the resulting cost matrix. However, it is still asymmetric. Note that the CV is the same as with the range-based method when we replace V_{task} by the CV of the first uniform law, $\frac{\sqrt{12}}{6}\frac{R_{\text{task}}-1}{R_{\text{task}}+1}$, and V_{mach} by the CV of the second uniform law, $\frac{\sqrt{12}}{6}\frac{R_{\text{mach}}-1}{R_{\text{mach}}+1}$.

3.3 Consistency Extension

Both the previous methods produce cost matrices that may not be representative of realistic settings. For instance, the costs of a given task is not correlated to the costs of another task, which may often be the case in practice. The consistency extension consists in reordering the costs in the generated matrix to have an instance that is closer to the uniform case. Specifically, the rows of a submatrix of an rows and bm columns are sorted. Thus, a machine that is faster for a given task than another machine will likely be also faster for another task. Inconsistent matrices have $a = b = 0$ while consistent matrices have $a = b = 1$ (other matrices are either called semiconsistent or partially consistent).

3.4 Usage in the Literature

We covered the English articles that cite at least one of the references in which the methods were initially presented and that were freely available. For each reference, we extracted all the distinct sets of parameters. Additionally, we differentiated between example cost matrices that illustrate the generation methods from cost matrices that are used in actual sets of experiments to study scheduling algorithms. However, the size was ignored as we only consider asymptotic properties (the impact of the size is assessed in [11, Section 4.6]).

Some data were not specifically provided. The parameters that could be directly inferred from the article or from similar works are emphasized: this concerns mostly missing parameters for the consistency extension (the ones from the cited article were taken). Otherwise, they are treated as missing values (denoted by NA). Some articles lack enough information, which prevented any parameter extraction.

On the 160 analyzed articles, 78 provide exploitable information on the cost matrix instances. The rest consists of 40 articles with no description, but which refer to instances described in other articles and 42 articles with unclear descriptions or approaches that do not fit the current study.

Table 2: Summary of the cost matrix properties with the range-based method. Asymptotic values are when both R_{task} and R_{mach} are large.

Property	Value
Expected value	$\frac{1}{4}(R_{\text{task}}+1)(R_{\text{mach}}+1)$
Standard deviation	$\frac{1}{12}\sqrt{(R_{\text{task}}-1)^2(R_{\text{mach}}-1)^2+3(R_{\text{mach}}-1)^2(R_{\text{task}}+1)^2+3(R_{\text{task}}-1)^2(R_{\text{mach}}+1)^2}$
CV	$\frac{1}{3}\sqrt{\frac{(R_{\text{task}}-1)^2(R_{\text{mach}}-1)^2}{(R_{\text{task}}+1)^2(R_{\text{mach}}+1)^2}+3\frac{(R_{\text{task}}-1)^2}{(R_{\text{task}}+1)^2}+3\frac{(R_{\text{mach}}-1)^2}{(R_{\text{mach}}+1)^2}}$
Distribution	Product of two uniform laws
Asymptotic expected value	$\frac{1}{4}R_{\text{task}}R_{\text{mach}}$
Asymptotic standard deviation	$\frac{\sqrt{7}}{12}R_{\text{task}}R_{\text{mach}}$
Asymptotic CV	$\frac{\sqrt{7}}{3}\approx 0.88$

Table 3: Summary of the cost matrix properties with the CVB method.

Property	Value
Expected value	μ_{task}
CV	$\sqrt{V_{\text{task}}^2V_{\text{mach}}^2+V_{\text{task}}^2+V_{\text{mach}}^2}$
Distribution	Product of two gamma laws

The extracted data are available in [10], [11, Appendix B] and summarized below. While most articles fail to precisely describe the used method, only the range and CV parameters are crucial for reproducing similar instances. In the end, 342 sets of parameters were extracted in 78 articles for a total of 210 unique settings: 37 for the range-based method and 173 for the CVB one.

Figure 1 depicts the values used with both methods. Although there is no clear agreement on which precise parameters are the most relevant, there are some common tendencies. Values for low heterogeneity are usually 10 and 100 for the range-based method and .1, .25 and .3 for the CVB method. Values for high heterogeneity are usually 100, 1e3, 3e3 and 1e5 for the range-based method and .3, .35, .4, .5, .6, .7, .9, 1 and 2 for the CVB method.

4 Heterogeneity Measures

Assessing the impact of heterogeneity on heuristic performance requires a method for quantifying the heterogeneity of the generated cost matrices.

4.1 TDH and MPH

The closest related measures are the TDH (Task Difficulty Homogeneity) and the MPH (Machine Performance Homogeneity) [16], [17]. The TDH computation is described in Algorithm 3. The value $TD[i]$ represents the difficulty of task i, namely whether it has small costs. After the ordering, the final sum computes the average ratio between similar tasks in terms of difficulty (which lies in the interval $(0,1]$). If this average is one, then tasks are all similar. If it is close to zero, then the task heterogeneity is large.

The MPH computation is analogous except that the sum on Line 2 is performed on each row instead of each column. This results in a measure of the machine heterogeneity.

These measures have three major shortcomings (as mentioned in Section 2). First, they are not intuitive (they require to invert costs, to order sums and to average ratios). Also, they do not rely on classical statistical measures, which makes deriving formal results more difficult. In particular, the ordering on Line 4 complicates formal analysis. A last notable problem is that the resulting values depend on the size of the matrix. In particular, it is close to one when the matrix is large (even if it is generated with the same parameters and has, intuitively, the same characteristics). For instance, if we consider only one machine, the following matrices (cost vectors in this case) have the same TDH: $[1,2]$ and $[0.125, 0.25, 0.5, 1, 2, 4]$. The second vector, however, seems more heterogeneous. As another example, let the minimum TD be 1 and the maximum TD be 100. Given Proposition 3, the TDH is always greater than 0.60 when there are 10 tasks and it is always greater than 0.95 when there are 100 tasks. This measure is thus relevant only for comparing small cost matrices with similar sizes.

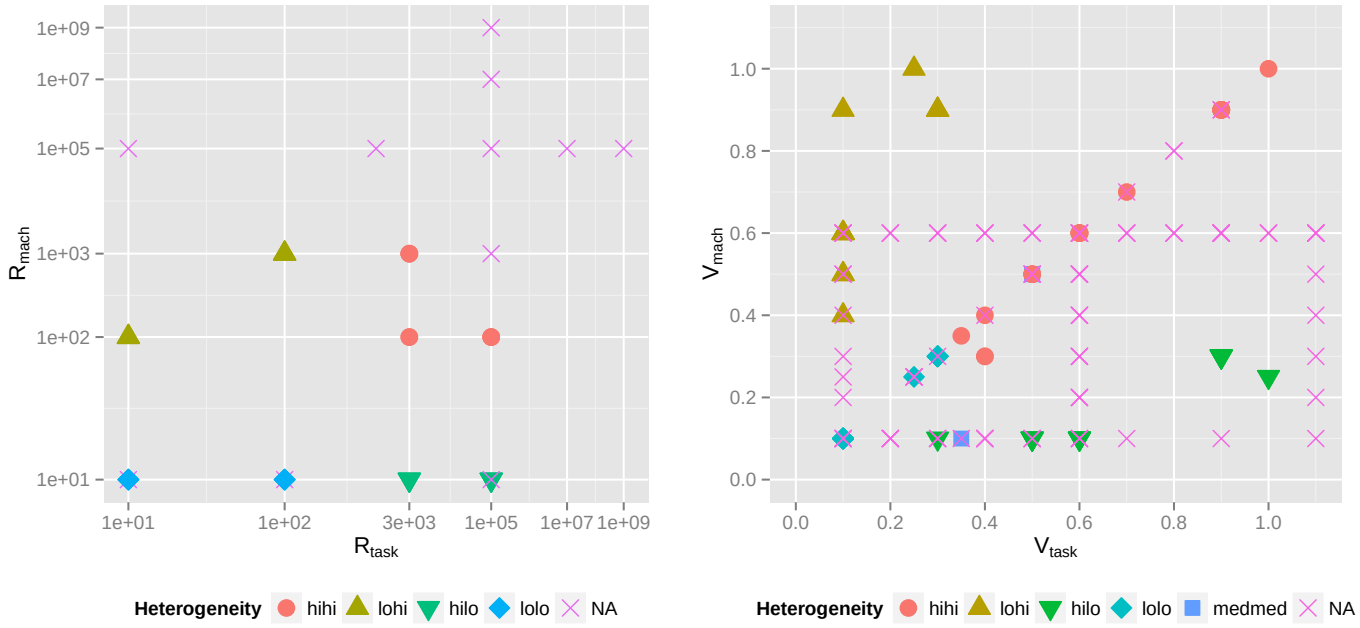
ALGORITHM 3: TDH computation

Input: a $n \times m$ cost matrix
Output: the TDH of this matrix
1: **for all** $1 \leq i \leq n$ **do**
2: $\quad TD[i] \leftarrow \sum_{j=1}^{m} \frac{1}{e_{i,j}}$
3: **end for**
4: sort TD in ascending order
5: **return** $\frac{1}{n-1}\sum_{i=1}^{n-1}\frac{TD[i]}{TD[i+1]}$

Proposition 3. *The TDH cannot be lower than* $e^{\log\left(\frac{\min(TD)}{\max(TD)}\right)/(n-1)}$.

Proof. The minimum TDH is achieved when the sum $\sum_{i=1}^{n-1}\frac{a_i}{a_{i+1}}$ where $a_i = TD[i]$ is minimum. Let $f : [a_1, a_n]^{n-2} \to (0,\infty)$ be the corresponding multivariate function with a_1 and a_n being constant. Each value a_i for $1 < i < n$ is greater than or equal to a_1 because the a_i are ordered. As a_1 represents an average cost and is thus strictly greater than zero, all nominators and all denominators are strictly greater than zero. Therefore, f is a continuous function from the compact $[a_1, a_n]^{n-2}$. The extreme value theorem states that a continuous function from a non-empty compact space to a subset of the real numbers attains a maximum and a minimum. This proves the existence of a minimum.

We now show by contradiction that this minimum is achieved when the ratios $\frac{a_i}{a_{i+1}}$ are all equal for $1 \leq i < n$. Assume it is not the case and let i be the lowest value for



(a) Range-based method

(b) CVB method

Figure 1: Parameters used in the literature. Three points are not shown for the CVB method: $(1.4, 0.4)$, $(1.8, 0.4)$ and $(0.1, 2)$.

which $\frac{a_i}{a_{i+1}} \neq \frac{a_{i+1}}{a_{i+2}}$, which can be rewritten as $a_{i+1} \neq \sqrt{a_i a_{i+2}}$. A lower value is attained when $a_{i+1} = \sqrt{a_i a_{i+2}}$ because the partial derivate of f with respect to a_{i+1} (i.e., $-\frac{a_{i-1}}{a_i^2} + \frac{1}{a_{i+1}}$) is zero with this value. Therefore, the minimum is achieved when all ratios $\frac{a_i}{a_{i+1}}$ are equal. This is the case when $a_i = e^{\log(TD[1]) + \frac{i-1}{n-1}\log\left(\frac{a_n}{a_1}\right)}$ for $1 \leq i \leq n$.

When replacing a_i by $TD[i]$, the TDH simplifies as $e^{\log\left(\frac{TD[1]}{TD[n]}\right)/(n-1)}$ or $e^{\log\left(\frac{\min(TD)}{\max(TD)}\right)/(n-1)}$ if the vector TD is not sorted. □

4.2 Intuitive Measures of Heterogeneity

We propose below two intuitive measures of task and machine heterogeneity that rely on classic properties:

- Assuming that the mean of each row represents a task weight, the task heterogeneity may be defined as the CV (Coefficient of Variation) of the means of the rows (noted $V\mu_{\text{task}}$). Analogously, the machine heterogeneity may be measured as the CV of the means of the columns (noted $V\mu_{\text{mach}}$).

$$CV \begin{Bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{Bmatrix} \begin{pmatrix} e_{1,1} & \cdots & e_{1,m} \\ \vdots & \ddots & \vdots \\ e_{n,1} & \cdots & e_{n,m} \end{pmatrix}$$

- Alternatively, the CV of one column may represent the task heterogeneity for a given machine. Therefore, the mean of the CV of the columns may measure the task heterogeneity (noted μV_{task}) while the mean of the CV of the rows may measure the machine heterogeneity (noted μV_{mach}).

The first measure of task and machine heterogeneity has been criticized for small instances [17]. The MPH is argued to outperform the CV as it is less sensitive to outliers. In this situation, the CV can be replaced by the quartile coefficient of dispersion, which is a similar standard statistical measure but is more difficult to formally analyze. Finally, the decision trees in [15] suggest that varying this measure has an impact on the heuristic performance and is thus significant.

With both measures, it is possible to use the standard deviation instead of the CV. However, the CV provides a relative measure that is independent from the cost mean. If an absolute measure is deemed more meaningful, the proposed measures can be adapted by using the standard deviation.

4.3 Coherence with the Uniform Model

The previous measures do not only rely on intuition, they are also consistent with the expectation when we consider the uniform model. In this model, the cost of executing a task i on a machine j is given by the product of the task weight, w_i, and the machine cycle time, b_j. The concept of task and machine heterogeneity is easy to grasp in the uniform model: it is given by the statistical dispersion of the weights and the speeds, respectively. We assume that the CV of the weights, noted CV_{task}, is a relevant measure of the task heterogeneity. Analogously, the CV of the speeds, noted CV_{mach}, represents the machine heterogeneity.

It is possible to convert an instance of the uniform model to the unrelated model because this last model is more general. The cost matrix is generated by combining both vectors $\{w_i\}_{1\leq i\leq n}$ and $\{b_j\}_{1\leq j\leq m}$ such that $e_{i,j} = w_i b_j$. As we know the heterogeneity properties of a uniform instance, we expect our proposed measures for the unrelated model to be consistent when applied on the converted instance.

Proposition 4. *Let $U = (\{w_i\}_{1\leq i\leq n}, \{b_j\}_{1\leq j\leq m})$ be a uniform instance and $E = \{e_{i,j}\}_{1\leq i\leq n, 1\leq j\leq m}$ be the corre-*

sponding unrelated instance such that $e_{i,j} = w_i b_j$. Then, $CV_{task}(U) = V\mu_{task}(E) = \mu V_{task}(E)$ and $CV_{mach}(U) = V\mu_{mach}(E) = \mu V_{mach}(E)$.

Proof. By definition, $CV_{\text{task}}(U) = \frac{\sqrt{\sum_{i=1}^{n} w_i^2/n - (\sum_{i=1}^{n} w_i/n)^2}}{\sum_{i=1}^{n} w_i/n}$ whereas $V\mu_{\text{task}}(E)$ is the CV of the means of the rows. The mean of row i is $\sum_{j=1}^{m} e_{i,j}/m = w_i/m \sum_{j=1}^{m} b_j$. Then, $V\mu_{\text{task}}(E) = \frac{\sqrt{\sum_{i=1}^{n} (w_i\phi)^2/n - (\sum_{i=1}^{n} w_i\phi/n)^2}}{\sum_{i=1}^{n} w_i\phi/n}$ where $\phi = \sum_{j=1}^{m} b_j/m$ is the mean of the inverse speeds. Therefore, $V\mu_{\text{task}}(E) = CV_{\text{task}}(U)$.

Remember that $\mu V_{\text{task}}(E)$ is the mean of the CV of the columns. The CV of column j, CV_j is

$$\begin{aligned} CV_j &= \frac{\sqrt{\sum_{i=1}^{n} e_{i,j}^2/n - (\sum_{i=1}^{n} e_{i,j}/n)^2}}{\sum_{i=1}^{n} e_{i,j}/n} \\ &= \frac{\sqrt{\sum_{i=1}^{n} (w_i b_j)^2/n - (\sum_{i=1}^{n} (w_i b_j)/n)^2}}{\sum_{i=1}^{n} (w_i b_j)/n} \\ &= CV_{\text{task}}(U) \end{aligned}$$

The mean of these CV is thus also $CV_{\text{task}}(U)$.

The demonstration is analogous for the machine heterogeneity measures. □

Proposition 4 shows that our proposed measures are consistent with the intuition on uniform instances.

4.4 Heterogeneity of the Range-Based and CVB Methods

We analyze the asymptotic heterogeneity properties of the CVB method with the proposed measures depending on the parameters V_{task} and V_{mach}. An estimator T converges to θ when the expected value of T tends to θ as the number of samples (n and m in our case) tends to ∞.

Proposition 5. *The measure $V\mu_{task}$ of a cost matrix generated using the CVB method with the parameters V_{task} and V_{mach} converges to V_{task} as $n \to \infty$ and $m \to \infty$.*

Proof. This proof assumes that the mean of a set of n samples (called the sample mean) of a random variable with mean μ and standard deviation σ is a random variable with mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$. Moreover, the CV of a set of n samples (called the sample CV) of a random variable with CV V converges to V as $n \to \infty$.

Let μ_i be the sample mean of the costs on row i. This row is the product of $q[i]$, which is a random variable that follows a distribution with mean μ_{task} and CV V_{task}, and m values that follow a distribution with mean one and CV V_{mach}. μ_i is thus also the product of the first random variable and the sample mean of the other m values, which follows a random variable with mean one and CV $\frac{V_{\text{mach}}}{\sqrt{m}}$. Therefore, the mean of μ_i is μ_{task} and its CV is $\sqrt{V_{\text{task}}^2 \frac{V_{\text{mach}}^2}{m} + \frac{V_{\text{mach}}^2}{m} + V_{\text{task}}^2}$, which tends to V_{task} as $m \to \infty$. The consistency properties have no impact on μ_i because only values on the same row are ordered. □

Proposition 6. *The measure $V\mu_{mach}$ of a cost matrix generated using the CVB method with the parameters V_{task} and V_{mach} converges to $a\sqrt{b}V_{mach}$ as $n \to \infty$ and $m \to \infty$.*

Proof. Let μ_j be the sample mean of the costs on column j. The measure $V\mu_{\text{mach}}$ is the ratio of the sample standard deviation of all μ_j, noted $\sigma\mu_{\text{mach}}$, to the sample mean of all μ_j, noted $\mu\mu_{\text{mach}}$.

Let's distinguish the columns where the costs are consistent ($1 \leq j \leq bm$) from the inconsistent columns ($bm < j \leq m$). For the inconsistent columns, μ_j is the sample mean of n values that follow a product between a distribution with mean μ_{task} and CV V_{task}, and a distribution with mean one and CV V_{mach}. Thus, μ_j follows a distribution with mean μ_{task} and CV $\sqrt{\frac{V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2}{n}}$ for $bm < j \leq m$. Therefore, the sample mean of μ_j converges to μ_{task} and its sample standard deviation converges to zero as $n \to \infty$ for $bm < j \leq m$.

For the consistent columns, $a \times n$ rows are sorted. Let q_p denotes the p-quantile of a distribution with mean one and CV V_{mach} (it is the value x for which $F(x) = p$ where F is the cumulative distribution function). Note that $e_{i,j} \to q[i]q_{j/(bm)}$ as $m \to \infty$ for $1 \leq i \leq an$ and $1 \leq j \leq bm$. Therefore, μ_j can be decomposed as a weighted sum of sample means (one for the sorted rows and another for the last rows): the first sample mean follows a distribution with mean $\mu_{\text{task}} q_{j/(bm)}$ and CV $\frac{V_{\text{task}}}{\sqrt{an}}$ while the second follows a distribution with mean μ_{task} and CV $\sqrt{\frac{V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2}{(1-a)n}}$. Therefore, the sample mean of μ_j converges to $a\mu_{\text{task}} q_{j/(bm)} + (1-a)\mu_{\text{task}}$ and its sample standard deviation converges to zero as $n \to \infty$ for $1 \leq j \leq bm$.

On one hand, $\mu\mu_{\text{mach}} = \frac{1}{m}\sum_{j=1}^{m} \mu_j = \frac{1}{m}(\sum_{j=1}^{bm}(a\mu_{\text{task}} q_{j/(bm)} + (1-a)\mu_{\text{task}}) + (1-b)m\mu_{\text{task}}) = ab\mu_{\text{task}} \frac{1}{bm}\sum_{j=1}^{bm} q_{j/(bm)} + (1-a)b\mu_{\text{task}} + (1-b)\mu_{\text{task}}$ as $n \to \infty$. Note that $\frac{1}{bm}\sum_{j=1}^{bm} q_{j/(bm)} = \int_0^1 q_p dp = 1$ as $m \to \infty$. Thus, $\mu\mu_{\text{mach}} = \mu_{\text{task}}$ as $n \to \infty$ and $m \to \infty$. On the other hand, as $n \to \infty$ and $m \to \infty$:

$$\begin{aligned} \sigma\mu_{\text{mach}} &= \sqrt{\frac{1}{m}\sum_{j=1}^{m} \mu_j^2 - \left(\frac{1}{m}\sum_{j=1}^{m} \mu_j\right)^2} \\ &= \sqrt{\frac{1}{m}\sum_{j=1}^{bm} \mu_j^2 + \frac{1}{m}\sum_{j=bm+1}^{m} \mu_j^2 - \mu_{\text{task}}^2} \\ &= \sqrt{\frac{1}{m}\sum_{j=1}^{bm} (a\mu_{\text{task}} q_{j/(bm)} + (1-a)\mu_{\text{task}})^2 + \overline{(1-b)\mu_{\text{task}}^2 - \mu_{\text{task}}^2}} \\ &= \mu_{\text{task}} \sqrt{\frac{1}{m}\sum_{j=1}^{bm} (a^2 q_{j/(bm)}^2 + 2a q_{j/(bm)}(1-a) + \overline{(1-a)^2) - b}} \end{aligned}$$

$$\begin{aligned} &= \mu_{\text{task}} \sqrt{a^2 b \frac{1}{bm}\sum_{j=1}^{bm} q_{j/(bm)}^2 + 2a(1-a)b\frac{1}{bm}\sum_{j=1}^{bm} q_{j/(bm)} + \overline{(1-a)^2 b - b}} \\ &= a\sqrt{b}\mu_{\text{task}} \sqrt{\frac{1}{bm}\sum_{j=1}^{bm} q_{j/(bm)}^2 - 1} \end{aligned}$$

Note that $\frac{1}{bm}\sum_{j=1}^{bm} q_{j/(bm)}^2 = \int_0^1 q_p^2 dp = \int_{-\infty}^{\infty} x^2 f(x) dx = V_{\text{mach}}^2 + 1$ as $m \to \infty$ with the substitution $p = F(x)$ and

Table 4: Summary of the heterogeneity properties of the CVB method.

Measure	Value
$V\mu_{\text{task}}$	V_{task}
μV_{task}	$\begin{cases}\Phi = \sqrt{V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2} & \text{if } a = 0 \\ bV_{\text{task}} + (1-b)\Phi & \text{if } a = 1\end{cases}$
$V\mu_{\text{mach}}$	$a\sqrt{b}V_{\text{mach}}$
μV_{mach}	V_{mach}

$dp = f(x)dx$ where f is the probability density function of a distribution with mean one and CV V_{mach}. This requires the distribution to be continuous, which is the case for the gamma distribution. Therefore, $\sigma\mu_{\text{mach}} = a\sqrt{b}\mu_{\text{task}}V_{\text{mach}}$ and $V\mu_{\text{mach}} = a\sqrt{b}V_{\text{mach}}$ as $n \to \infty$ and $m \to \infty$. □

Proposition 7. *The measure μV_{task} of a cost matrix generated using the CVB method with the parameters V_{task} and V_{mach} converges to $\sqrt{V_{task}^2 V_{mach}^2 + V_{task}^2 + V_{mach}^2}$ as $n \to \infty$ if the matrix is inconsistent and to $bV_{task} + (1-b)\sqrt{V_{task}^2 V_{mach}^2 + V_{task}^2 + V_{mach}^2}$ as $n \to \infty$ and $m \to \infty$ if $a = 1$.*

Proof. Let V_j be the sample CV of column j. When $a = 0$, the values on column j follow a distribution that is the product of a distribution with mean μ_{task} and CV V_{task}, and a distribution with mean one and CV V_{mach}. Therefore, V_j converges to $\sqrt{V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2}$ as $n \to \infty$. Since this value does not depends on j, μV_{task} (the sample mean of these sample CV) also converges to $\sqrt{V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2}$ as $n \to \infty$.

When $a = 1$, V_j still converges to $\sqrt{V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2}$ as $n \to \infty$ for $bm < j \leq m$. However, μ_j (the sample mean of column j) converges to $\mu_{\text{task}}q_{j/(bm)}$ as $n \to \infty$ and $m \to \infty$ while σ_j (the sample standard deviation of column j) converges to $\mu_{\text{task}}V_{\text{task}}q_{j/(bm)}$ as $n \to \infty$ and $m \to \infty$ for $1 \leq j \leq bm$. Thus, V_j converges to V_{task} as $n \to \infty$ and $m \to \infty$ for $1 \leq j \leq bm$. Therefore, μV_{task} converges to $bCV_{\text{task}} + (1-b)\sqrt{V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2}$ as $n \to \infty$ and $m \to \infty$. □

Proposition 8. *The measure μV_{mach} of a cost matrix generated using the CVB method with the parameters V_{task} and V_{mach} converges to V_{mach} as $m \to \infty$.*

Proof. Let V_i be the sample CV of row i. The values on row i follow a distribution that is the product of $q[i]$ and a distribution with mean one and CV V_{mach}. Therefore, V_i converges to V_{mach} as $m \to \infty$. Since this value does not depend on i, μV_{mach} (the sample mean of these sample CV) also converges to V_{mach} as $m \to \infty$. □

Table 4 synthesises the previous formal results. They can be extended to the range-based method by replacing V_{task} by the CV of the first random variable ($\frac{\sqrt{12}}{6}\frac{R_{\text{task}}-1}{R_{\text{task}}+1}$) and V_{mach} by the CV of the second one ($\frac{\sqrt{12}}{6}\frac{R_{\text{mach}}-1}{R_{\text{mach}}+1}$). Indeed, the proofs only use the mean and the CV of the underlying distributions. Moreover, the uniform distribution is also continuous. Although the formal analysis of μV_{task} for arbitrary values of a was unsuccessful, the following formula provides a close estimate: $a^2bV_{\text{task}} + (1-a^2b)\sqrt{V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2}$.

In the case of complete consistency (i.e., when $a = b = 1$), $V\mu_{\text{task}} = \mu V_{\text{task}} = V_{\text{task}}$ and $V\mu_{\text{mach}} = \mu V_{\text{mach}} = V_{\text{mach}}$, which supports the proposed heterogeneity measures. This special case is due to the fact that consistent cost matrices are closer to uniform instances than inconsistent ones and both measures are equivalent for uniform instances.

However, the CVB method has two issues. As a consequence of the asymmetry of the generation method, the task heterogeneity is not symmetric to the machine heterogeneity. For instance, we have $\mu V_{\text{task}} = \sqrt{V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2}$, whereas $V\mu_{\text{mach}} = V_{\text{mach}}$ for inconsistent matrices. This makes the generation method less direct as the parameters must be chosen such as to circumvent this asymmetry. In particular, if a high machine heterogeneity is required, then the task heterogeneity will also be high.

The second issue is related to the impact of the consistency parameters on the heterogeneity properties. It biases comparisons of scheduling methods when cost matrices are used with different consistency settings because these matrices will also have different heterogeneity properties. The range-based method presents an even stronger bias as both V_{task} and V_{mach} tend to $\frac{\sqrt{12}}{6}$ as $R_{\text{task}} \to \infty$ and $R_{\text{mach}} \to \infty$ (the heterogeneity properties are thus often similar).

4.5 Task and Machine Heterogeneity in Previous Studies

For each of the instances summarized in Section 3, we computed both heterogeneity measures using the formulas of Table 4 and the input parameters: R_{task} and R_{mach} for the range-based method; V_{task} and V_{mach} for the CVB method; and the consistency parameters, a and b, for both methods. For the case when $0 < a < 1$, μV_{task} was measured on a single 1000×1000 cost matrix that was generated with the range-based or the CVB method. When the consistency values are missing, matrices are assumed to be inconsistent. Finally, the mean is set to 1 when it is not given with the CVB method because it has no impact on any measure.

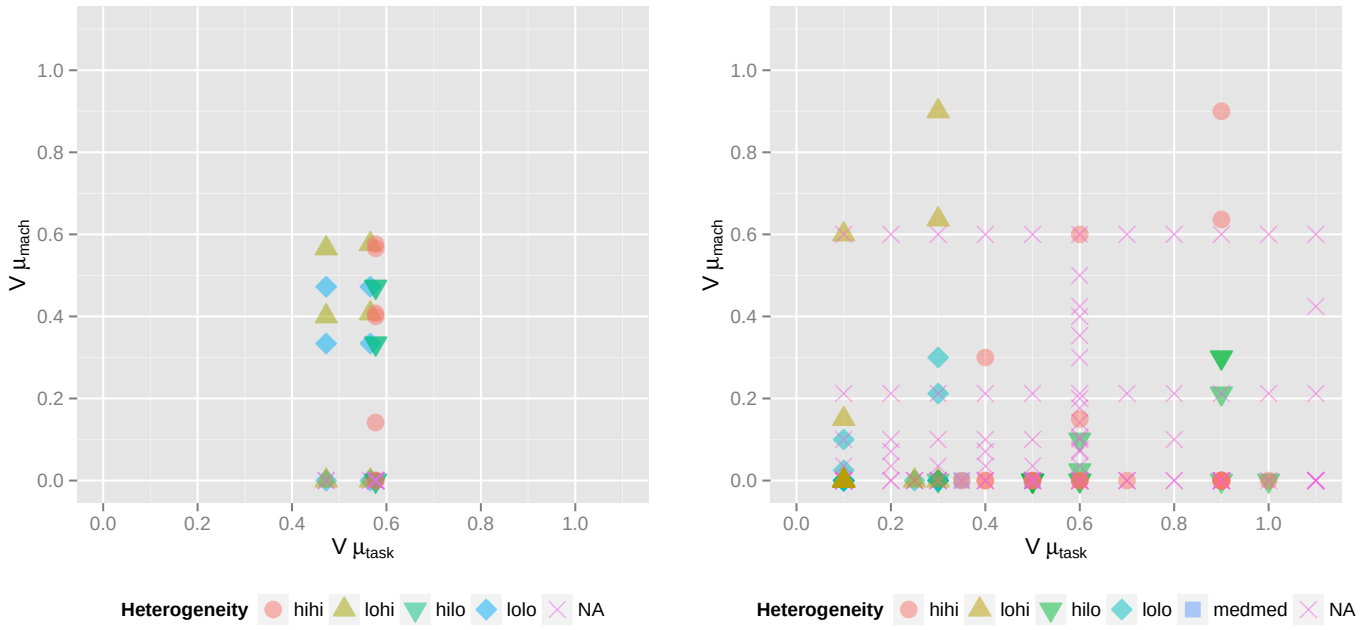
Figures 2 and 3 depict the values for the measures proposed above. The range-based method has a clear bias because many heterogeneity values have never been obtained. Also, the consistency parameters invalidate the claimed properties of the cost matrices with respect to the heterogeneity quadrant for both heterogeneity measures: some *hihi* instances have the same machine heterogeneity as *lolo* instances on Figure 2, whereas some *lohi* instances have the same task heterogeneity as *hilo* instances on Figure 3.

This analysis is also consistent with the observation made in [16] about the fact that the range-based and CVB methods do not cover the entire range of possible values for the MPH.

As mentioned in Section 4.2, both proposed heterogeneity measures are relative. This allows a direct comparison between each heterogeneity value. Using the standard deviation instead would require normalizing them for this analysis.

5 Controlling the Heterogeneity

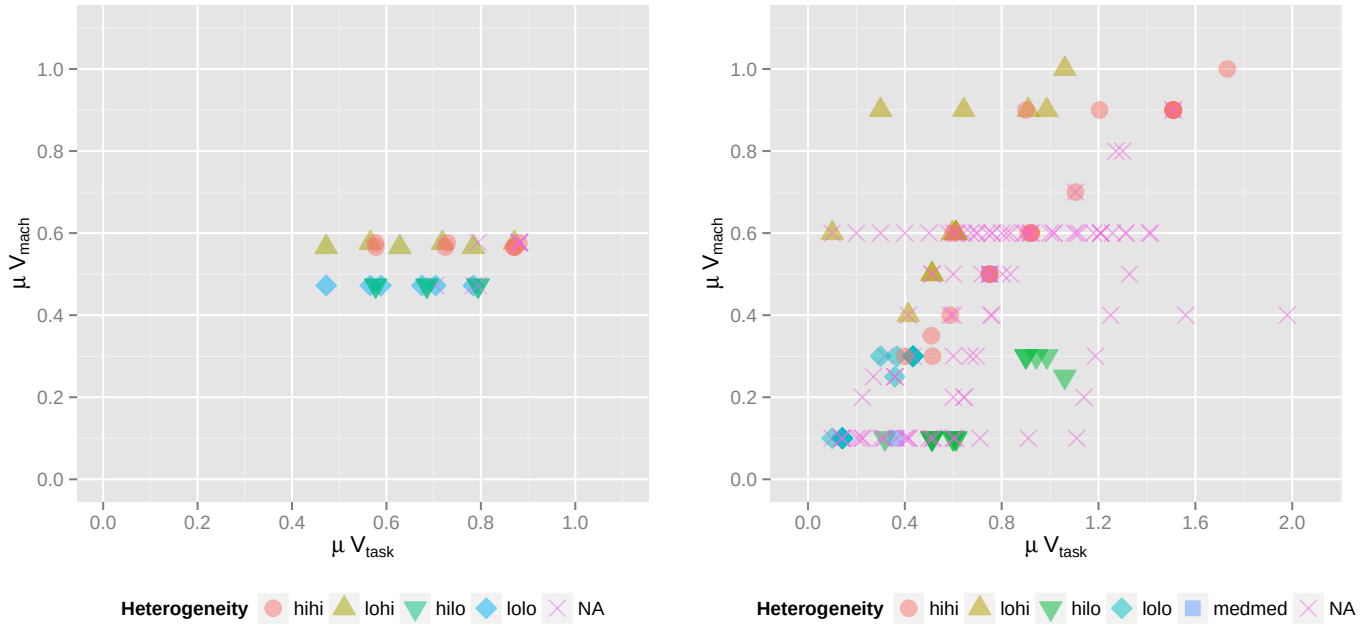
We are interested in generating cost matrices that have specific heterogeneity properties according to the measures introduced in Section 4. We propose two methods that both alter a cost matrix generated from uniform instances for which we control the task and machine heterogeneities. These cost matrices have specific properties in terms of consistency and correlation between each row and each column, and the proposed methods introduce some randomness in it. They both possess the same time complexity (i.e., $O(nm)$).



(a) Range-based method

(b) CVB method

Figure 2: Heterogeneity properties ($V\mu_{\text{task}}$ and $V\mu_{\text{mach}}$) of cost matrices used in the literature. Two points are not shown for the CVB method: $(1.4, 0)$ and $(1.8, 0)$.



(a) Range-based method

(b) CVB method

Figure 3: Heterogeneity properties (μV_{task} and μV_{mach}) of cost matrices used in the literature. The x-scale is twice as large as in Figure 2 for the CVB method because large values of V_{mach} tends to increase the measure μV_{task}. One point is not shown for the CVB method: $(2.01, 2)$.

5.1 Shuffling Method

The first proposed method shuffles the costs in the matrix that corresponds to a uniform instance (see Algorithm 4). It first generates the task weights on Line 2 and the inverse of the machine speeds on Line 5. The corresponding matrix is computed on Line 9 before starting the shuffling part. For each cost $e_{i,j}$, another cost $e_{i',j'}$ is selected on a different row and column (Lines 14 and 15). The same amount is then removed from these costs and is added to two other costs, $e_{i,j'}$ and $e_{i',j}$ (one that is on the same row as the first cost and on the same column as the second, and another one that is on the same row as the second cost and on the same column as the first). This step (Lines 25 to 28) preserves the mean of each row and the mean of each column. The heterogeneity properties thus remain the same.

The transferred amount is the largest value (in absolute) such that no cost among the four considered costs becomes lower than the minimum one among them (this prevents costs to be arbitrarily low). For instance, if $e_{i,j}$ is the minimum cost (i.e., $e_{i,j} = \min(e_{i,j}, e_{i',j}, e_{i,j'}, e_{i',j'})$), there are two cases: if $e_{i,j'} < e_{i',j}$, then $e_{i,j'}$ becomes the new minimum and the added value to $e_{i,j}$ and to $e_{i',j'}$ is $e_{i,j'} - e_{i,j}$; otherwise, it is $e_{i',j} - e_{i,j}$.

Maintaining both the minimum and the maximum cost is not possible because the cost matrix is generated from a uniform instance. This method focuses only on preventing costs to be arbitrarily low because it is critical to guarantee positive costs.

Proposition 9. *When used with parameters V_{task} and V_{mach}, the shuffling method generates costs with expected value* 1*.*

Proof. Costs in the matrix corresponding to the uniform matrix follow a distribution that is the product of two distributions with mean one. Therefore, the expected value of the costs in the matrix before the shuffling step is also one. The shuffling step does not change the expected value of the costs because the amount that is taken on any cost is given to another cost. □

Proposition 10. *The measure $V\mu_{task}$ of a cost matrix generated using the shuffling method with the parameters V_{task} and V_{mach} converges to V_{task} as $n \to \infty$.*

Proof. Analogously to the proof of Proposition 9, the shuffling step has no impact on the mean of each row and each column. The measure $V\mu_{\text{task}}$ is thus the same for the final cost matrix as for the intermediate matrix that corresponds to a uniform instance.

As a corollary of Proposition 4, the sample CV of the sample means of all rows in this intermediate matrix is equal to the sample CV of the vector $\{w_i\}_{1 \leq i \leq n}$. This last sample CV converges to V_{task} as $n \to \infty$. □

Proposition 11. *The measure $V\mu_{mach}$ of a cost matrix generated using the shuffling method with the parameters V_{task} and V_{mach} converges to V_{mach} as $m \to \infty$.*

Proof. Due to the symmetry of the shuffling method, the proof is analogous to the proof of Proposition 10. □

Table 5 summarizes the formal results related to the shuffling method.

ALGORITHM 4: Shuffling cost matrix generation with gamma distribution

Input: n, m, V_{task}, V_{mach}
Output: a $n \times m$ cost matrix

1: **for all** $1 \leq i \leq n$ **do**
2: $\quad w_i \leftarrow G(1/V_{\text{task}}^2, V_{\text{task}}^2)$
3: **end for**
4: **for all** $1 \leq j \leq m$ **do**
5: $\quad b_j \leftarrow G(1/V_{\text{mach}}^2, V_{\text{mach}}^2)$
6: **end for**
7: **for all** $1 \leq i \leq n$ **do**
8: $\quad$ **for all** $1 \leq j \leq m$ **do**
9: $\quad\quad e_{i,j} \leftarrow w_i b_j$
10: $\quad$ **end for**
11: **end for**
12: **for all** $1 \leq i \leq n$ **do**
13: $\quad$ **for all** $1 \leq j \leq m$ **do**
14: $\quad\quad i' \leftarrow (U(1, n-1) + i - 1 \mod n) + 1$
15: $\quad\quad j' \leftarrow (U(1, n-1) + j - 1 \mod m) + 1$
16: $\quad\quad$ **if** $e_{i,j} = \min(e_{i,j}, e_{i',j}, e_{i,j'}, e_{i',j'})$ **then**
17: $\quad\quad\quad d \leftarrow \min(e_{i',j} - e_{i,j}, e_{i,j'} - e_{i,j})$
18: $\quad\quad$ **else if** $e_{i',j} = \min(e_{i',j}, e_{i,j'}, e_{i',j'})$ **then**
19: $\quad\quad\quad d \leftarrow -\min(e_{i,j} - e_{i',j}, e_{i',j'} - e_{i',j})$
20: $\quad\quad$ **else if** $e_{i,j'} = \min(e_{i,j'}, e_{i',j'})$ **then**
21: $\quad\quad\quad d \leftarrow -\min(e_{i,j} - e_{i,j'}, e_{i',j'} - e_{i,j'})$
22: $\quad\quad$ **else**
23: $\quad\quad\quad d \leftarrow \min(e_{i',j} - e_{i',j'}, e_{i,j'} - e_{i',j'})$
24: $\quad\quad$ **end if**
25: $\quad\quad e_{i,j} \leftarrow e_{i,j} + d$
26: $\quad\quad e_{i',j} \leftarrow e_{i',j} - d$
27: $\quad\quad e_{i,j'} \leftarrow e_{i,j'} - d$
28: $\quad\quad e_{i',j'} \leftarrow e_{i',j'} + d$
29: $\quad$ **end for**
30: **end for**
31: **return** $\{e_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$

Table 5: Summary of the cost matrix properties with the shuffling method.

Property	Value
Expected value	1
$V\mu_{\text{task}}$	V_{task}
$V\mu_{\text{mach}}$	V_{mach}

5.2 Noise-Based Method

The second method, described in Algorithm 5, relies on a simple idea, which was also used in [19]: each cost of a matrix, which corresponds to a uniform instance, is multiplied by a random variable with mean one (Line 9).

Proposition 12. *When used with parameters V_{task}, V_{mach} and V_{noise}, the noise-based method generates costs with expected value one and CV $\frac{\sqrt{V_{task}^2 V_{mach}^2 V_{noise}^2 + V_{task}^2 V_{mach}^2 + V_{task}^2 V_{noise}^2 + V_{mach}^2 V_{noise}^2}}{+V_{task}^2 + V_{mach}^2 + V_{noise}^2}$.*

Proof. Each cost is the product of three random variables that have all the same mean one. Additionally, their CV (and standard deviations in this case) are V_{task}, V_{mach} and V_{noise}. The global CV can be derived by remarking that the CV of

ALGORITHM 5: Noise-based cost matrix generation with gamma distribution

Input: n, m, V_{task}, V_{mach}, V_{noise}
Output: a $n \times m$ cost matrix
for all $1 \leq i \leq n$ **do**
  $w_i \leftarrow G(1/V_{\text{task}}^2, V_{\text{task}}^2)$
end for
for all $1 \leq j \leq m$ **do**
  $b_j \leftarrow G(1/V_{\text{mach}}^2, V_{\text{mach}}^2)$
end for
for all $1 \leq i \leq n$ **do**
  for all $1 \leq j \leq m$ **do**
    $e_{i,j} \leftarrow w_i b_j \times G(1/V_{\text{noise}}^2, V_{\text{noise}}^2)$
  end for
end for
return $\{e_{i,j}\}_{1 \leq i \leq n, 1 \leq j \leq m}$

the product of two random variables with CV V_1 and V_2 is $\sqrt{V_1^2 V_2^2 + V_1^2 + V_2^2}$. □

Proposition 13. *The measure $V\mu_{task}$ of a cost matrix generated using the noise-based method with the parameters V_{task}, V_{mach} and V_{noise} converges to V_{task} as $n \to \infty$ and $m \to \infty$.*

Proof. Let μ_i be the sample mean of row i. This row is the product of w_i, which follows a distribution with mean one and CV V_{task}, and m values that are each the product of a random variable with mean one and CV V_{mach} and a random variable with mean one and CV V_{noise}. μ_i is thus also the product of w_i and the sample mean of the other m values, which follows a random variable with mean one and CV $\sqrt{\frac{V_{\text{mach}}^2 V_{\text{noise}}^2 + V_{\text{mach}}^2 + V_{\text{noise}}^2}{m}}$. Therefore, the mean of μ_i is one and its CV is $\sqrt{V_{\text{task}}^2 \frac{V_{\text{mach}}^2 V_{\text{noise}}^2 + V_{\text{mach}}^2 + V_{\text{noise}}^2}{m} + V_{\text{task}}^2 + \frac{V_{\text{mach}}^2 V_{\text{noise}}^2 + V_{\text{mach}}^2 + V_{\text{noise}}^2}{m}}$, which tends to V_{task} as $m \to \infty$. Therefore, the sample CV of all μ_i converges to V_{task} as $n \to \infty$ and $m \to \infty$. □

Proposition 14. *The measure $V\mu_{mach}$ of a cost matrix generated using the noise-based method with the parameters V_{task}, V_{mach} and V_{noise} converges to V_{mach} as $n \to \infty$ and $m \to \infty$.*

Proof. Due to the symmetry of the noise-based method, the proof is analogous to the proof of Proposition 13. □

Proposition 15. *The measure μV_{task} of a cost matrix generated using the noise-based method with the parameters V_{task}, V_{mach} and V_{noise} converges to $\sqrt{V_{task}^2 V_{noise}^2 + V_{task}^2 + V_{noise}^2}$ as $n \to \infty$.*

Proof. Let V_j be the sample CV of column j. Each column is the product of b_j and n values that are each the product of a random variable with mean one and CV V_{task} and a random variable with mean one and CV V_{noise}. Thus, V_j converges to the CV of this product (i.e., $\sqrt{V_{\text{task}}^2 V_{\text{noise}}^2 + V_{\text{task}}^2 + V_{\text{noise}}^2}$) as $n \to \infty$. Therefore, the measure μV_{task} converges to $\sqrt{V_{\text{task}}^2 V_{\text{noise}}^2 + V_{\text{task}}^2 + V_{\text{noise}}^2}$ as $n \to \infty$. □

Proposition 16. *The measure μV_{mach} of a cost matrix generated using the noise-based method with the parameters V_{task}, V_{mach} and V_{noise} converges to $\sqrt{V_{mach}^2 V_{noise}^2 + V_{mach}^2 + V_{noise}^2}$ as $m \to \infty$.*

Table 6: Summary of the cost matrix properties with the noise-based method.

Property	Value
Expected value	1
CV	$\sqrt{V_{\text{task}}^2 V_{\text{mach}}^2 V_{\text{noise}}^2 + V_{\text{task}}^2 V_{\text{mach}}^2 + V_{\text{task}}^2 V_{\text{noise}}^2 + V_{\text{mach}}^2 V_{\text{noise}}^2 + V_{\text{task}}^2 + V_{\text{mach}}^2 + V_{\text{noise}}^2}$
Distribution	Product of three gamma laws
$V\mu_{\text{task}}$	V_{task}
μV_{task}	$\sqrt{V_{\text{task}}^2 V_{\text{noise}}^2 + V_{\text{task}}^2 + V_{\text{noise}}^2}$
$V\mu_{\text{mach}}$	V_{mach}
μV_{mach}	$\sqrt{V_{\text{mach}}^2 V_{\text{noise}}^2 + V_{\text{mach}}^2 + V_{\text{noise}}^2}$

Proof. Due to the symmetry of the noise-based method, the proof is analogous to the proof of Proposition 15. □

Table 6 summarizes the formal results related to the noise-based method.

This method requires one additional parameter: V_{noise}. When the objective is to have cost matrices with specific values of $V\mu_{\text{task}}$ and $V\mu_{\text{mach}}$ (for large n and m), we propose to set V_{noise} to $\min(V_{\text{task}}, V_{\text{mach}})$. This limits the amount of noise in the costs.

Contrary to the shuffling method, the noise-based method can also generate cost matrices with specific values of μV_{task} and μV_{mach} (asymptotically). The parameters can be fixed as follow: if $\mu V_{\text{task}} < \mu V_{\text{mach}}$, then $V_{\text{task}} = 0$, $V_{\text{noise}} = \mu V_{\text{task}}$ and $V_{\text{mach}} = \sqrt{(\mu V_{\text{mach}}^2 - \mu V_{\text{task}}^2)/(\mu V_{\text{task}}^2 + 1)}$; otherwise, $V_{\text{mach}} = 0$, $V_{\text{noise}} = \mu V_{\text{mach}}$ and $V_{\text{task}} = \sqrt{(\mu V_{\text{task}}^2 - \mu V_{\text{mach}}^2)/(\mu V_{\text{mach}}^2 + 1)}$. This setting maximizes the amount of noise.

Even though the shuffling method has less formal results (probably due to its combinatoric operations), the noise-based method has two drawbacks: the additional parameter is not trivial to determine and the method introduces more variation in the costs than the shuffling method. This makes this method more complex to use.

6 Impact on Scheduling Heuristics

This section assesses the impact of the heterogeneity properties defined in Section 4 on the relative performance of some classic heuristics.

6.1 Scheduling Heuristics

Our intention here is not to find the best heuristic but rather to show the impact of the cost matrix generation method on the performance results. We use classical heuristics from the literature summarized in Table 7. Most of them (OLB, MET, MCT, Min-min, Max-min, HEFT, HLPT, Suff) are list-based algorithms. The Genetic Algorithm (GA) relies on an initial population containing a solution obtained with Min-min. In addition to these classic heuristics, we added two more elaborated methods (the Bal prefixed methods) that try to reconsider an initial mapping obtained from MET (Minimum Execution Time) mapping: any task is moved to the machine that will finish it the earliest if it does not increase

Table 7: Summary of the scheduling heuristics for the $R||C_{\max}$ problem.

Name	Ref	Complexity	Remark
OLB	[4]	nm	Opportunistic Load Balancing
MET	[4]	nm	Minimum Execution Time
MCT	[4]	nm	Minimum Completion Time
Min-min	[4]	n^2m	Earliest finish time of smallest task
Max-min	[4]	n^2m	Earliest finish time of largest task
Suff	[23]	n^2m	Task that will suffer most first
GA	[4]	–	Genetic Algorithm
HEFT	[24]	$nm + n\log(n)$	Heterogeneous Earliest-Finish-Time
HLPT	[25]	$nm + n\log(n)$	Heterogeneous version of LPT
GreedySuff		$nm\log(m)$	Greedy allocation based on sufferage
BalSuff		–	Reconsider MET mapping
BalEFT		–	Reconsider MET mapping

the maximum completion time. These heuristics are described in [11, Appendix C].

Getting relevant reference values (lower bounds on the makespan) for our performance measures is not straightforward in practice due to the heterogeneity of the problem. We thus rely on a variation of the genetic algorithm to provide an estimation of these values. The initial population is initialized, in addition to other random individuals, with all the solutions obtained by the other algorithms. The population evolution is based on the algorithm description given in [4]. An elite chromosome is maintained so that the resulting solution cannot be worse than any of the initial solutions and thus the genetic algorithm is no worse than any of the other algorithms.

6.2 Settings

Cost matrices are generated with three different methods: the shuffling method and the noise-based method with two approaches to set the noise (see Section 5.2). In all cases, there are two parameters: $V\mu_{\text{task}}$ and $V\mu_{proc}$ for the first two methods and μV_{task} and μV_{proc} for the last one. These two parameters are distributed in the range $[0.001, 10]$ with 30 equidistant values using a probit scale (i.e., 0.001, 0.0014, 0.0019, 0.0026, ..., 5.3, 7.3, 10).

All methods rely on the gamma distribution. However, when the CV is close to 10, it may generate zero values (it occurs in 1.4% of all the generated costs) due to rounding. The resulting cost matrix is altered to avoid this by setting to 2.225074e-308 (this is the value of the smallest non-zero normalized floating-point number, `double.xmin`, in R 3.3.0) each zero value. Otherwise some tasks may have no weight, which requires specific handling and is not realistic. The impact is however marginal and concerns only matrices for which $V_{task} > 2.8$ or $V_{mach} > 2.8$.

For each pair of parameters, 200 cost matrices are generated with $n = 100$ tasks and $m = 30$ machines. For each scenario, we compute the makespan of each heuristic. We only consider the relative difference from the reference makespan: $|C - C_{\min}|/C_{\min}$ where C is the makespan of a given heuristic and $C_{\min}$ is the best makespan we obtained (the genetic algorithm initialized with all the solutions obtained by the other heuristics). The closer to zero, the better the performance.

6.3 Results

Figure 4 contains heat maps of the relative performance for each algorithm. On each figure, we use a logarithmic scale on both axes: the x-axis gives the heterogeneity value for the tasks ($V\mu_{\text{task}}$ or μV_{task}) while the y-axis gives the heterogeneity value for the machines ($V\mu_{\text{mach}}$ or μV_{mach}). The bottom-left area represents almost homogeneous instances (same cost for each execution) while the top-right area is the most heterogeneous one. The heterogeneity values covered by the range-based and CVB methods in the literature are represented with dark rectangles on each sub-figure.

The scales on each heat map start at 0.001. We consider that an heterogeneity that is below this value is negligible and that a heuristic that is closer to the reference makespan than this value is good enough. For instance, BalSuff may be considered near-optimal when the heterogeneity values are below 1%.

Figure 4 uses the shuffling method with the heterogeneity measure $V\mu_{\text{task}}/V\mu_{\text{mach}}$. Similar figures can be obtained with the noise-based method using either of the heterogeneity measures [11, Figures 6 and 7].

Figure 5 plots the best heuristic depending on the heterogeneity properties. Contour lines show the number of heuristics which performance is closer to the best heuristics than 0.001. For instance, there are at least 5 heuristics whose relative performances are almost equivalent when task heterogeneity is high (i.e., if the best heuristic average relative difference from the reference value is 0.004, then at least 5 other heuristics have a relative difference lower than 0.005).

The heuristics are ordered by the number of instances for which no other heuristic produces a better solution. When several heuristics are equivalent for a given tile, the appearing heuristic is the one that is the best the least often. This allows one to see even the settings for which the worst heuristics may be good.

6.4 Analysis

The settings cover a large part of the possible instances for the $R||C_{\max}$ problem. Specific scheduling problems may be associated to some areas on the figures. Problems considering homogeneous (*i.e.*, identical) tasks are situated on the left area: $P|p_i = p|C_{\max}$ (*i.e.*, same machine speeds) in the bottom corner and $Q|p_i = p|C_{\max}$ (*i.e.*, distinct machine speeds) for the above part. Inversely, problems considering tasks with varying weights allocated to homogeneous machines, the $P||C_{\max}$ problem, are situated on the bottom area. While the first two problems can be solved in polynomial time, the last problem is NP-complete.

The heat maps suggest that the area where the heterogeneity values are between 0.1 and 1 is more challenging for most heuristics (areas in dark blue on the heat maps are 30% far from the reference). This is confirmed by Figure 5 where there is often a single best heuristic with these settings. Oppositely, many heuristics are close to the best one when the task heterogeneity is low or high, or when the machine heterogeneity is high. On one hand, execution costs are similar

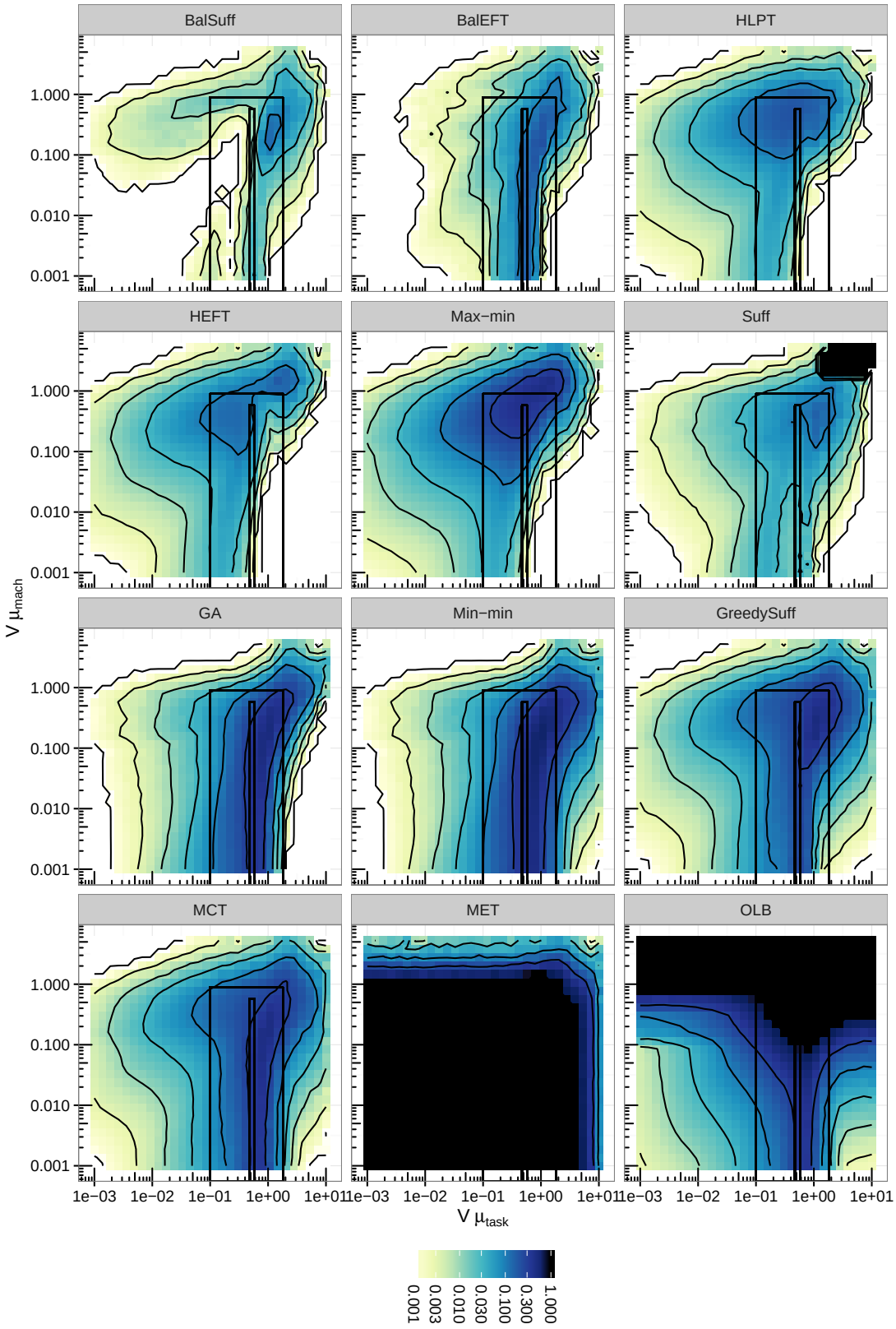


Figure 4: Heuristic performance relatively to the best case with the shuffling method. Values below 0.001 are white and values above 1 are black. Contour lines correspond to the levels in the legend (0.001, 0.003, . . .). The rectangles correspond to the properties covered by the range-based and CVB methods in the literature (see Figure 2).

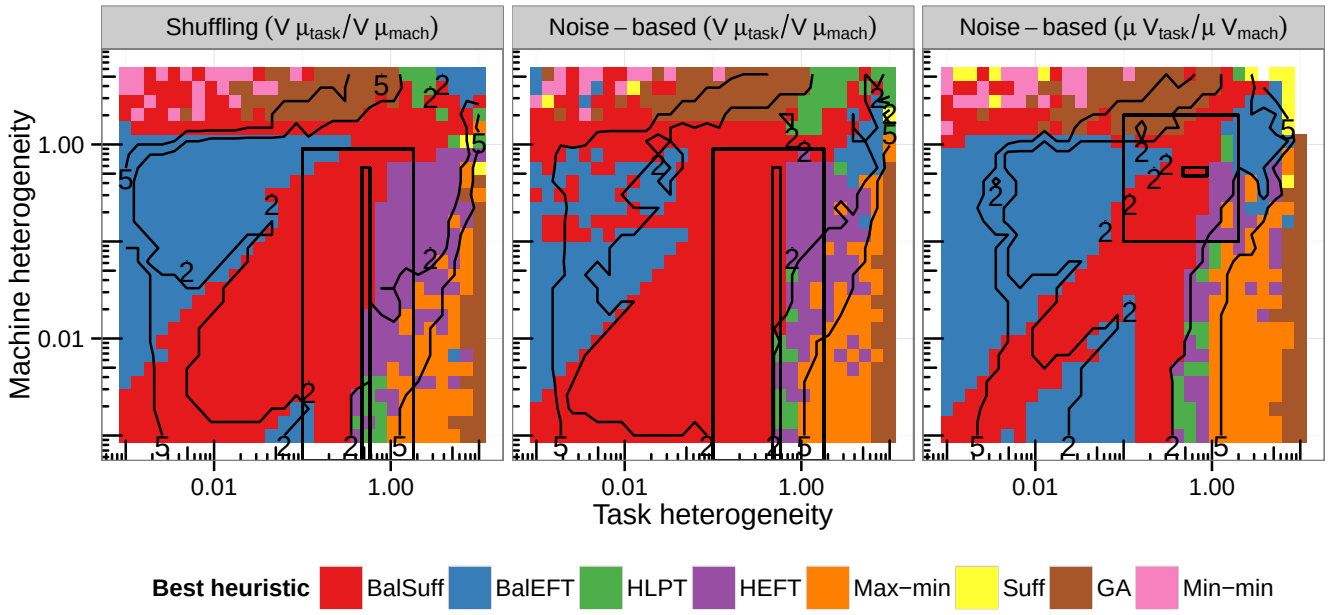


Figure 5: Best heuristic in the average case with the shuffling and the noise-based method with $V\mu_{\text{task}}$ and $V\mu_{\text{mach}}$ or μV_{task} and μV_{mach} as parameters. Contour lines correspond to the number of heuristics with a performance closer to the best heuristic performance than 0.001. The dark rectangles correspond to the properties covered by the range-based and CVB methods in the literature (see Figures 2 and 3).

when the coefficient of variation is below 0.1. A non-optimal allocation will thus have a lower impact than with higher heterogeneity. On the other hand, most execution costs are close to zero when the coefficient of variation is higher than 1 and bad allocations may be easy to avoid because there are few allocations that are extremely critical while most of them are not. It is thus easier to generate a reasonable schedule.

When the machine heterogeneity is low (with medium task heterogeneity), there is often a single best heuristic. This suggests that these settings leads to difficult instances. As mentioned above, this is close to the $P||C_{\max}$ problem. We may conclude that dealing with heterogeneous tasks is more difficult than with heterogeneous machines, which is also supported by the asymmetry of the heat maps.

Finally, Figure 5 shows the best heuristics: BalSuff when both heterogeneity properties are comparable, BalEFT when the machine heterogeneity is higher than the task heterogeneity and HEFT/HLPT when the task heterogeneity is high.

Overall, we used two generation methods and two heterogeneity measures (one with the shuffling method and two with the noise-based method) and this analysis stands in all cases.

The range-based and CVB generation methods used in the literature could not provide these results due to two factors: the heterogeneity properties of the generated instances have a limited coverage (shown by the dark rectangles) and the erroneous claimed properties of these matrices prevent an unbiased analysis.

6.5 Discussion

This study focuses on the impact of some measures (either $V\mu_{\text{task}}$ and $V\mu_{\text{mach}}$, or μV_{task} and μV_{mach}) on the performance of twelve heuristics. However, other properties could be measured. If we consider the skewness and the kurtosis as in [15], we can think of 4×4 measures for the rows and as many for the columns. The main limitation of this study is to ignore the effect of all these possible measures. In addition, this study cannot be directly extended to assess all their possible interactions.

Another limitation is related to the effect of outliers. For large instances, the law of large number applies and the measures proposed in Section 4 correspond to the characteristics of the cost matrices. However, for small instances, we suggest switching to robust measures such as the median, the interquartile range and the quartile coefficient of dispersion instead of the mean, the standard deviation and the CV, respectively.

7 Conclusion

This study shows that the methods used in the literature for generating cost matrices are biased: the claimed heterogeneity properties of these instances are invalidated by the two measures we proposed to quantify them. We also show that the range of instances that has been used are restricted. It is specifically the case for the range-based method that covers only a minor fraction of all the possible settings in terms of heterogeneity. By providing new cost matrix generation methods, we show that heuristics for the $R||C_{\max}$ problem have interesting behavior outside this restriction. For instance, BalEFT is the best heuristic when the task heterogeneity is low and this could not have been shown with the instances used in the literature. Overall, this study provides tools to help the assessment of scheduling strategies.

In addition to all the possible measures mentioned in Section 6.5, we plan to analyze other properties, in particular the correlation. It would also be interesting to see if the conclusions hold for some variations of the $R||C_{\max}$ problem such as considering arrival times or online scheduling.

Acknowledgments

We would like to thank Pierre-Cyrille Héam for his helpful comments on the proof of Proposition 3.

We also sincerely thank the reviewers for their careful reading and detailed comments.

Computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté.

References

[1] J. Y.-T. Leung, Ed., *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Chapman & Hall/CCR, 2004.

[2] A. Saltelli, K. Chan, and E. M. Scott, *Sensitivity analysis*. Wiley New York, 2009.

[3] A. K. Bardsiri and S. M. Hashemi, "A Comparative Study on Seven Static Mapping Heuristics for Grid Scheduling Problem," *International Journal of Software Engineering and Its Applications*, vol. 6, no. 4, pp. 247–256, 2012.

[4] T. D. Braun, H. J. Siegel, N. Beck, L. L. Bölöni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao, D. Hensgen, and R. F. Freund, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," *Journal of Parallel and Distributed Computing*, vol. 61, no. 6, pp. 810–837, 2001.

[5] C. O. Diaz, J. E. Pecero, and P. Bouvry, "Scalable, low complexity, and fast greedy scheduling heuristics for highly heterogeneous distributed computing systems," *The Journal of Supercomputing*, vol. 67, no. 3, pp. 837–853, 2014.

[6] P. Luo, K. Lü, and Z. Shi, "A revisit of fast greedy heuristics for mapping a class of independent tasks onto heterogeneous computing systems," *Journal of Parallel and Distributed Computing*, vol. 67, no. 6, pp. 695–714, 2007.

[7] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[8] S. Ali, H. J. Siegel, M. Maheswaran, D. Hensgen, and S. Ali, "Representing task and machine heterogeneities for heterogeneous computing systems," *Tamkang Journal of Science and Engineering*, vol. 3, no. 3, pp. 195–208, 2000.

[9] S. Ali, H. J. Siegel, M. Maheswaran, and D. Hensgen, "Task execution time modeling for heterogeneous computing systems," in *Heterogeneous Computing Workshop (HCW)*. IEEE, 2000, pp. 185–199.

[10] L.-C. Canon and L. Philippe, "Code for On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms," *http://dx.doi.org/10.6084/m9.figshare.1321295.v3*, Mar. 2015.

[11] ——, "On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms," FEMTO-ST, Tech. Rep. RR-FEMTO-ST-8663, Mar. 2015.

[12] ——, "On the Heterogeneity Bias of Cost Matrices when Assessing Scheduling Algorithms," in *Euro-Par*, 2015, pp. 109–121.

[13] R. K. Armstrong Jr, "Investigation of effect of different run-time distributions on SmartNet performance," DTIC Document, Tech. Rep., 1997.

[14] S. Ali, "A comparative study of dynamic mapping heuristics for a class of independent tasks onto heterogeneous computing systems," Ph.D. dissertation, Purdue University, 1999.

[15] A. M. Al-Qawasmeh, A. A. Maciejewski, H. Wang, J. Smith, H. J. Siegel, and J. Potter, "Statistical measures for quantifying task and machine heterogeneities," *The Journal of Supercomputing*, vol. 57, no. 1, pp. 34–50, 2011.

[16] A. M. Al-Qawasmeh, A. A. Maciejewski, and H. J. Siegel, "Characterizing heterogeneous computing environments using singular value decomposition," in *International Parallel & Distributed Processing Symposium Workshops and Phd Forum (IPDPSW)*. IEEE, 2010, pp. 1–9.

[17] A. M. Al-Qawasmeh, A. A. Maciejewski, R. G. Roberts, and H. J. Siegel, "Characterizing task-machine affinity in heterogeneous computing environments," in *International Parallel & Distributed Processing Symposium Workshops and Phd Forum (IPDPSW)*. IEEE, 2011, pp. 34–44.

[18] R. Friese, B. Khemka, A. A. Maciejewski, H. J. Siegel, G. A. Koenig, S. Powers, M. Hilton, J. Rambharos, G. Okonski, and S. W. Poole, "An analysis framework for investigating the trade-offs between system performance and energy consumption in a heterogeneous computing environment," in *International Parallel & Distributed Processing Symposium Workshops and Phd Forum (IPDPSW)*. IEEE, 2013, pp. 19–30.

[19] A. M. Al-Qawasmeh, S. Pasricha, A. A. Maciejewski, and H. J. Siegel, "Power and Thermal-Aware Workload Allocation in Heterogeneous Data Centers," *Transactions on Computers*, vol. 64, no. 2, pp. 477–491, 2013.

[20] B. Khemka, R. Friese, S. Pasricha, A. A. Maciejewski, H. J. Siegel, G. A. Koenig, S. Powers, M. Hilton, R. Rambharos, and S. Poole, "Utility maximizing dynamic resource management in an oversubscribed energy-constrained heterogeneous computing system," *Sustainable Computing: Informatics and Systems*, vol. 5, pp. 14–30, 2014.

[21] S. Ghosh and S. G. Henderson, "Behavior of the NORTA method for correlated random vector generation as the dimension increases," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 13, no. 3, pp. 276–294, 2003.

[22] M. Cryan, M. Dyer, L. A. Goldberg, M. Jerrum, and R. Martin, "Rapidly mixing markov chains for sampling contingency tables with a constant number of rows," *SIAM Journal on Computing*, vol. 36, no. 1, pp. 247–278, 2006.

[23] H. Casanova, A. Legrand, D. Zagorodnov, and F. Berman, "Heuristics for Scheduling Parameter Sweep Applications in Grid Environments." in *Heterogeneous Computing Workshop (HCW)*. IEEE, 2000, pp. 349–363.

[24] H. Topcuoglu, S. Hariri, and M.-y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE transactions on parallel and distributed systems*, vol. 13, no. 3, pp. 260–274, 2002.

[25] R. L. Graham, "Bounds on Multiprocessing Timing Anomalies," *Journal of Applied Mathematics*, vol. 17, no. 2, pp. 416–429, 1969.

Louis-Claude Canon received the master degree in computer science in 2007 from both the ESEO (Ecole Supérieure d'Electronique de l'Ouest) and the University of Angers. He started the PhD degree on uncertainty management in parallel systems at the Loria (Laboratoire Lorrain de Recherche en Informatique et ses Applications) and finalized it at the LaBRI (Laboratoire Bordelais de Recherche en Informatique). He received the PhD degree in computer science from the University of Nancy in 2010. After his PhD, he spent two years as a postdoc: one year at the University of Grenoble and one year at the Irisa laboratory in Rennes. He is an associate professor of computer science at the University of Franche-Comté and conducting research at FEMTO-ST. His main research interests include scheduling, stochastic optimization, and reproducible research.

Laurent Philippe obtained his Ph.D degree in 1993 from the University of Franche-Comté. He was an associate professor from 1993 to 2001 and since a full Professor of computer science at the University of Franche-Comté. He is leading a research team on distributed computing at FEMTO-ST Institut and he head the regional computing center. His main research interests include distributed systems, parallelism and scheduling.